



ATID Application Development Framework Reference Manual–RFID13.56Mhz

Revision: Ver. 0.2

Date: January, 2012

ATID Co.,Ltd

Table of Contents

Table of Contents	2
Acronym	4
Revision History	5
1 .NET API Reference	6
1.1 Enumerations	6
1.1.1 MSR_RESULT	6
1.2 Reply Words list.....	7
1.2.1 "ER00" or "None".	7
1.2.2 "ER02"	7
1.2.3 "ER03"	7
1.2.4 "ER04"	7
1.2.5 "OK"	7
1.3 Memory Map.....	8
1.3.1 Mifare	8
1.3.2 ISO 15693	9
1.4 Methods.....	11
1.4.1 Open	11
1.4.2 Close	11
1.4.3 IsOpen.....	11
1.4.4 AntennaOff.....	11
1.4.5 ReadFirmwareVersion	12
1.4.6 MifareReadUid	12
1.4.7 MifareReadBlock.....	12
1.4.8 MifareWriteBlock.....	13
1.4.9 MifareKeyLoad.....	13
1.4.10 MifareIncrement	14
1.4.11 MifareDecrement	15
1.4.12 MifareUltraReadBlock	16
1.4.13 MifareUltraWriteBlock	16
1.4.14 Iso15693ReadUid	17
1.4.15 Iso15693ReadBlock	17
1.4.16 Iso15693WriteBlock	18
1.4.17 Iso15693LockBlock	18
1.4.18 Iso15693ReadAfi.....	18
1.4.19 Iso15693WriteAfi	19
1.4.20 Iso15693LockAfi.....	19
1.4.21 Iso15693WriteDsfid	20

1.4.22	Iso15693LockDsfid.....	20
1.4.23	Iso14443APDU.....	20
2	C/C++ API Reference	22
2.1	Enumerations.....	22
2.1.1	MSR_RESULT.....	22
2.2	Reply Words list.....	23
2.2.1	"ER00" or "None"	23
2.2.2	"ER02"	23
2.2.3	"ER03"	23
2.2.4	"ER04"	23
2.2.5	"OK"	23
2.3	Memory Map.....	24
2.3.1	Mifare	24
2.3.2	ISO 15693	25
2.4	Methods.....	27
2.4.1	Rfid1356Open.....	27
2.4.2	Rfid1356Close.....	27
2.4.3	Rfid1356IsOpen	27
2.4.4	Rfid1356AntennaOff.....	27
2.4.5	Rfid1356ReadFirmwareVersion	28
2.4.6	Rfid1356MifareReadUid	28
2.4.7	Rfid1356MifareReadBlock	28
2.4.8	Rfid1356MifareWriteBlock.....	29
2.4.9	Rfid1356MifareKeyLoad	30
2.4.10	Rfid1356MifareIncrement.....	30
2.4.11	Rfid1356MifareDecrement	31
2.4.12	Rfid1356MifareUltraReadBlock	32
2.4.13	Rfid1356MifareUltraWriteBlock.....	32
2.4.14	Rfid1356Iso15693ReadUid	33
2.4.15	Rfid1356Iso15693ReadBlock.....	33
2.4.16	Rfid1356Iso15693WriteBlock.....	34
2.4.17	Rfid1356Iso15693LockBlock.....	34
2.4.18	Rfid1356Iso15693ReadAfi	34
2.4.19	Rfid1356Iso15693WriteAfi.....	35
2.4.20	Rfid1356Iso15693LockAfi	35
2.4.21	Rfid1356Iso15693WriteDsfid.....	36
2.4.22	Rfid1356Iso15693LockDsfid	36
2.4.23	Rfid1356Iso14443APDU	36

Acronym

modules	descriptions
AADF	ATID Application Development Framework
HF	High Frequency
AFI	Application Family Identifier
DSFID	Data Storage Format Identifier

Revision History

Version	Date	Reason	Description	Author
0.1	2012/01/17	Draft		Y. J. CHO
0.2	2013/01/28	Update	- Transfer Function of ISO 14443 APDU added.	Y. J. CHO

1 .NET API Reference

1.1 Enumerations

1.1.1 MSR_RESULT

The call result of function

- **HF_RESULT_SUCCESS**
Success of function execution.
- **HF_RESULT_INVALID_ARGS**
Invalid parameter.
- **HF_RESULT_OUTOFMEMORY**
Fail to allocate resources.
- **HF_RESULT_UNSUPPORTED**
Currently UnsupportedCommands
- **HF_RESULT_ALREADY_OPENED**
RFID HF device is already opened.
- **HF_RESULT_NOT_OPENED**
Calling function not opened.
- **HF_RESULT_FAILURE**
Fail to execute function.
- **HF_RESULT_INVALID_DEVICE**
MSR device is not equipped.

1.2 Reply Words list

1.2.1 “ER00” or “None”.

Tag not found.

1.2.2 “ER02”

Fail to read, Key error.

1.2.3 “ER03”

Fail to write.

1.2.4 “ER04”

Command error.

1.2.5 “OK”

Success of command execution.

1.3 Memory Map

1.3.1 Mifare

Sector	Block	Byte Number within a Block																Description
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
39	15	Key A				Access Bits				Key B								Sector Trailer 39
	14																	Data
	13																	Data

	2																	Data
	1																	Data
	0																	Data
..
..
..
32	15	Key A				Access Bits				Key B								Sector Trailer 32
	14																	Data
	13																	Data

	2																	Data
	1																	Data
	0																	Data
31	3	Key A				Access Bits				Key B								Sector Trailer 31
	2																	Data
	1																	Data
	0																	Data
..
..
..
0	3	Key A				Access Bits				Key B								Sector Trailer 0
	2																	Data
	1																	Data
	0																	Manufacturer Data

MifareStandard(4Kbyte) Card is composed of 256 16Byte blocks. Sector00 ~ Sector1F(32 Sector) consists of 4block each, Sector20~Sector27(8 Sector) consists of 16block each.

In sector, there is a part called Access Condition which defines access of Card.If recording incorrectly on that part, there is possible that that sectors becomes unavailable, so be careful.

In case of Mifare Standard, Access Condition is included in final black of each sector (this block also called Sector Trailer)

1.3.2 ISO 15693

Memory map of Tag-it HF-I

Block	Byte0	Byte1	Byte2	Byte3	LockBit
00	User Data	User Data	User Data	User Data	—
01	User Data	User Data	User Data	User Data	—
⋮	⋮	⋮	⋮	⋮	⋮
0F	User Data	User Data	User Data	User Data	—
10	User Data	User Data	User Data	User Data	—
⋮	⋮	⋮	⋮	⋮	⋮
3F	User Data	User Data	User Data	User Data	—

UID 0	UID 1	UID 2	UID 3	Lock
UID 4	UID 5	UID 6	UID 7	Lock
				DSFID
				AFI
IC Version				

Tag-it HF-I is composed of 64Blocks of 4Byte Block(in total, 256Byte) as User Data. LockBit applies to Write Access Condition of iCode I and iCode SLI, and it is Configured each block of User data.

Memory map of My-d SRF

태그의종류		Page	Byte No.								관리영역	
			0	1	2	3	4	5	6	7	미사용	엑세스 조건
55V10 P	55V02P	Service Area	00	UID							8	9
			01								55	46
			02								55	66
		User Area	03	User Data							55	AA
			04	User Data							55	AA
			⋮	⋮							⋮	⋮
	X	User Area	1F	User Data							55	AA
			20	User Data							55	AA
			⋮	⋮							⋮	⋮
			⋮	⋮							⋮	⋮
			7F	User Data							55	AA

Each page is composed of 8Byte of datapart and 2Byte of management domain (intatol, 10Byte), and whether or not to write on this page is registered in Access Condition of management domain.

Both Reading and Writing of My-d SRF are executed per 1 page(8Byte)

Memory map of MB89R118

	Block	8 Byte / 1 Block
User 영역	00	User Data
	01	User Data
	⋮	⋮
	F8	User Data
	F9	User Data
System 영역	FA	UID
	FB	그 외의 ID 및 EAS Status
	FC	Block Security Status #1(64bit)
	FD	Block Security Status #2(64bit)
	FE	Block Security Status #3(64bit)
	FF	Block Security Status #4(LSB 부터 58bit)

System 영역의 상세는 아래의 표와 같다.

Block	Bit64-57	56-49	48-41	40-33	32-25	24-17	16-9	8-1
FA	UID							
FB	EAS	미 사용			DSFID Lock 스테이트	AFI Lock 스테이트	DSFID	AFI
FC	3F~00 Block 의 Block Security Status (LSB:00 Block)							
FD	7F~40 Block 의 Block Security Status							
FE	BF~80 Block 의 Block Security Status							
FF	6bit 미 사용	F9~C0 Block 의 Block Security Status						

MB89R118 is composed of 256 Blocks of 8Byte(in total, 2048 byte), inner 250Blocks(2000byte) becomes User Data, and information whether or not to write about each block is registered in Block Security Status(250Bit).

1.4 Methods

1.4.1 Open

Allocating system sources and opening HF RFID device.

`HF_RESULT` Open();

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

1.4.2 Close

Deallocating system resources and closing HF RFID device.

`HF_RESULT` Close();

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

1.4.3 IsOpen

Checking whether HF RFID device is open

`BOOL`IsOpened();

Parameters

None

Return Values

Open Status of RFID device.

True:Opend, False:Closed

1.4.4 AntennaOff

Stopping RF oscillation of RFID Antenna compulsorily

`HF_RESULT`AntennaOff();

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

1.4.5 ReadFirmwareVersion

Reading Firmware version of RFID device.

```
HF_RESULT ReadFirmwareVersion(  
    String sVersion  
);
```

Parameters

sVersion

firmware version

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

1.4.6 MifareReadUid

Reading UID of mifare tag within the range of response

```
HF_RESULT MifareReadUid (  
    String sUid  
);
```

Parameters

sUid

mifare Tag의 UID

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

1.4.7 MifareReadBlock

Reading designated Memory Block of mifare tag within the range of response.

```
HF_RESULT MifareReadBlock (  
    String sBlock,  
    String sKeyType,  
    String sKeySet,  
    String sReadData
```

);

Parameters

sBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

sKeyType

access key. Ex) "A" or "B"

sKeySet

key set. Ex) "0" or "1", or "2"

sReadData

String variable that will designate block value.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

1.4.8 MifareWriteBlock

Recording data in designated Memory Block of mifaretag within the range of response.

HF_RESULTMifareWriteBlock (

String sBlock,

String sKeyType,

String sKeySet,

String sWriteData

);

Parameters

sBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

sKeyType

access key. Ex) "A" or "B"

sKeySet

key set. Ex) "0" or "1", or "2"

sWriteData

Data which will be stored in block.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

1.4.9 MifareKeyLoad

Record data in Memory Block of designated mifare tag within the range of response

```
HF_RESULTMifareKeyLoad (
    StringsBlock,
    String sKeyType,
    String sKeySet,
    String sKey
);
```

Parameters

sBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

sKeyType

access key. Ex) "A" or "B"

sKeySet

key set. Ex) "0" or "1", or "2"

sKey

Key which will be loaded in RFID device.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Load of key is sector-by-sector, in case of parameter szBlock you must designate random block included in the sector. As the initial value of authentication key, both AKey and BKey are "FFFFFFFFFFFFFFFF", even if the power is disconnected, the registered authentication key will be remembered.

1.4.10 MifareIncrement

Adding nValue(decimal) to Value Block value.

```
HF_RESULTMifareIncrement (
    StringsBlock,
    String sKeyType,
    String sKeySet,
    uintnValue
);
```

Parameters

sBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

sKeyType

access key. Ex) "A" or "B"

sKeySet

key set. Ex) "0" or "1", or "2"

nValue

Value to be added(decimal)

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Access is 1 sheet 1 block at one time, and Mifare Light cannot be added.(limitations of Card)

1.4.11 MifareDecrement

Subtracting nValue(decimal) from Value Block.

HF_RESULTMifareDecrement (

String sBlock,

String sKeyType,

String sKeySet,

uint nValue

);

Parameters

sBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

sKeyType

access key. Ex) "A" or "B"

sKeySet

key set. Ex) "0" or "1", or "2"

nValue

Value to be subtracted(decimal)

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Access is 1 sheet 1 block at one time,

1.4.12 MifareUltraReadBlock

Reading block of MifareUltra Light Tag within the range of response.

```
HF_RESULTMifareUltraReadBlock (
    StringsTagNum,
    String sBlock,
    String sReadNum,
    String sReadData
);
```

Parameters

sTagNum

Turn of tag to read. Must input "01"

sBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

sReadNum

Digits of data to read.

nReadData

String variable that will store block value.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

there will be fail answer from block number 16 on, and reading blocks are limited to a maximum of 16. sum of The Block Number and the number of block are limited to 16.

1.4.13 MifareUltraWriteBlock

Reading block of MifareUltra Light Tag within within the range of response

```
HF_RESULTMifareUltraWriteBlock (
    StringsTagNum,
    String sBlock,
    String sWriteData
);
```

Parameters

sTagNum

Turn of tag to read. Must input "01"

sBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

sWriteData

Data which will be stored in block.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

The block number "00", "01" are the section that writing is forbidden.

1.4.14 Iso15693ReadUid

Reading UID of ISO15693 Tag within the range of response.

```
HF_RESULT Iso15693ReadUid (  
    StringsUid  
);
```

Parameters

sUid

String variable where UID will be stored.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

1.4.15 Iso15693ReadBlock

Reading data of designated block of ISO15693 Tag within the range of response

```
HF_RESULT Iso15693ReadBlock (  
    StringsBlock,  
    String sReadCount,  
    String sReadData  
);
```

Parameters

sBlock

Designate block to read as hexadecimal number. Ex) "00" or "01" or "0F"

sReadCount

Designate the number of block to read as hexadecimalEx) "00" or "01" or "0F"

sReadData

String variable where block data will be stored.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

1.4.16 Iso15693WriteBlock

Recording data to designated block of ISO15693 Tag within the range of response

```
HF_RESULT Iso15693WriteBlock (
    StringsBlock,
    String sWriteData
);
```

Parameters

sBlock

Designate block to read as hexadecimal number. Ex) "00" or "01" or "0F"

sWriteData

Data which will record block.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Section which can record data are User Data, AFI and DSFID.

1.4.17 Iso15693LockBlock

Apply write-protection to designated block of ISO15693 Tag within the range of response.

```
HF_RESULT Iso15693LockBlock (
    StringsBlock
);
```

Parameters

sBlock

Designate block to read as hexadecimal number. Ex) "00" or "01" or "0F"

sWriteData

Data which will record block

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

data can lock 1 block(4Byte) at a time, once locked block cannot be reverted to writable status.

1.4.18 Iso15693ReadAfi

Reading UID of Tag at the same AFI(Application Family Identifier) status among ISO15693

Tag within the range of response

```
HF_RESULT Iso15693ReadAfi (  
    StringsData,  
    String sAfi  
);
```

Parameters

sData

String variable for saving UID.

sAfi

Application family identifier

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

Notes

AFI 1byte is composed of top 4bit of Family and sub 4bit of sub-Family.

If Sub-Family is 0, matching the Family is detected, and if both Family and sub-Family are 0, all tag are detected. Depending on type of tag, if family and sub-Family doesn't coincide, there is possibility of non-responding

1.4.19 Iso15693WriteAfi

Changing AFI(Application Family Identifier) registered in Tag

```
HF_RESULT Iso15693WriteAfi (  
    String sAfi  
);
```

Parameters

sAfi

AFI to be registered in Tag(hexadecimal)

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

1.4.20 Iso15693LockAfi

Apply write-protection to AFI(Application Family Identifier) registered.in Tag

```
HF_RESULT Iso15693LockAfi();
```

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Once locked AFI cannot be reverted to writable status.

1.4.21 Iso15693WriteDsfid

Changing DSFID(Data Storage Format Identifier) registered in Tag

```
HF_RESULT Iso15693WriteDsfid(
    StringsDsfid
);
```

Parameters

sDsfid

DSFID to be registered in Tag(hexadecimal)

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

1.4.22 Iso15693LockDsfid

Apply write-protection to DSFID(Data Storage Format Identifier) registered in Tag.

```
HF_RESULT Iso15693LockDsfid();
```

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

Notes

Once locked DSFID cannot be reverted to writable stuff

1.4.23 Iso14443APDU

Transmitting ISO 14443A/B APDU(application protocol data unit)

```
HF_RESULT Iso14443APDU(
    StringSendData,
    StringResultData
);
```

Parameters

SendData

APDU command which will be transmitted to Rfid Module.

ResultData

The result about transmitted APDU command

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

2 C/C++ API Reference

2.1 Enumerations

2.1.1 MSR_RESULT

Showing the calling result of function

- **HF_RESULT_SUCCESS**
success of function execution.
- **HF_RESULT_INVALID_ARGS**
invalid parameter
- **HF_RESULT_OUTOFMEMORY**
fail to allocate resources.
- **HF_RESULT_UNSUPPORTED**
not currently-supported command
- **HF_RESULT_ALREADY_OPENED**
RFID HF device is already opened.
- **HF_RESULT_NOT_OPENED**
Calling function not opened.
- **HF_RESULT_FAILURE**
Fail to execute function.
- **HF_RESULT_INVALID_DEVICE**
MSR device is not equipped.

2.2 Reply Words list

2.2.1 “ER00” or “None”

Tag not found.

2.2.2 “ER02”

Fail to read, Key error.

2.2.3 “ER03”

Fail to write.

2.2.4 “ER04”

Command error.

2.2.5 “OK”

Success of command execution.

2.3 Memory Map

2.3.1 Mifare

Sector	Block	Byte Number within a Block																Description
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
39	15	Key A				Access Bits				Key B								Sector Trailer 39
	14																	Data
	13																	Data

	2																	Data
	1																	Data
	0																	Data
32	15	Key A				Access Bits				Key B								Sector Trailer 32
	14																	Data
	13																	Data

	2																	Data
	1																	Data
	0																	Data
31	3	Key A				Access Bits				Key B								Sector Trailer 31
	2																	Data
	1																	Data
	0																	Data
0	3	Key A				Access Bits				Key B								Sector Trailer 0
	2																	Data
	1																	Data
	0																	Manufacturer Data

MifareStandard(4Kbyte) Card is composed of 256 16Byte blocks. Sector00 ~ Sector1F(32 Sector) consists of 4block each, Sector20~Sector27(8 Sector) consists of 16block each.

In sector, there is a part called Access Condition which defines access of Card. If recording incorrectly on that part, there is possible that that sectors(in case of Light, whole card) becomes unavailable, so be careful.

In case of Mifare Standard, Access Condition is included in final block of each sector (this block also called Sector Trailer)

2.3.2 ISO 15693

Memory map of Tag-it HF-I

Block	Byte0	Byte1	Byte2	Byte3	LockBit
00	User Data	User Data	User Data	User Data	—
01	User Data	User Data	User Data	User Data	—
⋮	⋮	⋮	⋮	⋮	⋮
0F	User Data	User Data	User Data	User Data	—
10	User Data	User Data	User Data	User Data	—
⋮	⋮	⋮	⋮	⋮	⋮
3F	User Data	User Data	User Data	User Data	—

UID 0	UID 1	UID 2	UID 3	Lock 뎀
UID 4	UID 5	UID 6	UID 7	Lock 뎀
				DSFID
				AFI
IC Version				

Tag-it HF-I is composed of 64Blocks of 4Byte Block(in total, 256Byte) as User Data. LockBit applies to Write Access Condition of iCode I and iCode SLI, and it is Configured each block of User data.

Memory map of My-d SRF

태그의종류		Page	Byte No.								관리영역	
			0	1	2	3	4	5	6	7	미사용	엑세스 컨디션
55V10 P	55V02P	Service Area	00	UID							8	9
			01								55	46
			02								55	66
		User Area	03	User Data							55	76
			04	User Data							55	AA
	X	User Area	⋮	⋮							⋮	⋮
			1F	User Data							55	AA
			20	User Data							55	AA
			⋮	⋮							⋮	⋮
			7F	User Data							55	AA

Each page is composed of 8Byte of datapart and 2Byte of management domain (intatol, 10Byte), and whether or not to write on this page is registered in Access Condition of management domain.

Both Reading and Writing of My-d SRF are executed per 1 page(8Byte)

Memory map of MB89R118

	Block	8 Byte / 1 Block
User 영역	00	User Data
	01	User Data
	⋮	⋮
	F8	User Data
	F9	User Data
System 영역	FA	UID
	FB	그 외의 ID 및 EAS Status
	FC	Block Security Status #1(64bit)
	FD	Block Security Status #2(64bit)
	FE	Block Security Status #3(64bit)
	FF	Block Security Status #4(LSB 부터 58bit)

System 영역의 상세는 아래의 표와 같다.

Block	Bit64~57	56~49	48~41	40~33	32~25	24~17	16~9	8~1
FA	UID							
FB	EAS	미 사용			DSFID Lock 스테이트	AFI Lock 스테이트	DSFID	AFI
FC	3F~00 Block 의 Block Security Status (LSB:00 Block)							
FD	7F~40 Block 의 Block Security Status							
FE	BF~80 Block 의 Block Security Status							
FF	6bit 미 사용	F9~C0 Block 의 Block Security Status						

MB89R118 is composed of 256 Blocks of 8Byte(in total, 2048 byte), inner 250Blocks(2000byte) becomes User Data, and information whether or not to write about each block is registered in Block Security Status(250Bit).

2.4 Methods

2.4.1 Rfid1356Open

Allocating system sources and opening HF RFID device.

```
HF_RESULT Rfid1356Open();
```

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

2.4.2 Rfid1356Close

Deallocating system resources and closing HF RFID device.

```
HF_RESULT Rfid1356Close();
```

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

2.4.3 Rfid1356IsOpen

Checking whether HF RFID device is open

```
BOOL Rfid1356IsOpened ();
```

Parameters

None

Return Values

Open Status of RFID device.

True:Open, False:Closed

2.4.4 Rfid1356AntennaOff

Stopping RF oscillation of RFID Antenna compulsorily

`HF_RESULTRfid1356AntennaOff ();`

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

2.4.5 Rfid1356ReadFirmwareVersion

Reading Firmware version of RFID device.

`HF_RESULTRfid1356ReadFirmwareVersion(
 LPWSTRszVersion
);`

Parameters

szVersion

firmware version

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

2.4.6 Rfid1356MifareReadUid

Reading UID of mifare tag within the range of response

`HF_RESULTRfid1356MifareReadUid (
 LPWSTRszUid
);`

Parameters

szUid

mifare Tag의 UID

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

2.4.7 Rfid1356MifareReadBlock

Reading designated Memory Block of mifare tag within the range of response.

`HF_RESULTRfid1356MifareReadBlock (
 LPWSTRszBlock,
 LPWSTRszKeyType,
 LPWSTRszKeySet,`

```
LPWSTRszReadData
);
```

Parameters

szBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

szKeyType

access key. Ex) "A" or "B"

szKeySet

key set. Ex) "0" or "1", or "2"

szReadData

Variable which will store block value.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

2.4.8 Rfid1356MifareWriteBlock

Recording data in designated Memory Block of mifare tag within the range of response.

```
HF_RESULTRfid1356MifareWriteBlock (
    LPWSTRszBlock,
    LPWSTRszKeyType,
    LPWSTRszKeySet,
    LPWSTRszWriteData
);
```

Parameters

szBlock

designating block as 16 hexadecimal number. Ex) "00" or "01" or "0F"

szKeyType

access key. Ex) "A" or "B"

szKeySet

key set. Ex) "0" or "1", or "2"

szWriteData

Data which will be stored in block

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

2.4.9 Rfid1356MifareKeyLoad

Recording data in Memory Block of designated mifare tag within the range of response

```
HF_RESULTRfid1356MifareKeyLoad (
    LPWSTRszBlock,
    LPWSTRszKeyType,
    LPWSTRszKeySet,
    LPWSTRszKey
);
```

Parameters

szBlock

designating block as 16 hexadecimal number. Ex) "00" or "01" or "0F"

szKeyType

access key. Ex) "A" or "B"

szKeySet

key set. Ex) "0" or "1", or "2"

szKey

Key which will be loaded in RFID device.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Load of key is sector-by-sector, in case of parameter *szBlock* you must designate random block included in the sector. As the initial value of authentication key, both AKey and BKey are "FFFFFFFFFFFFFFFF", even if the power is disconnected, the registered authentication key will be remembered.

2.4.10 Rfid1356MifareIncrement

Adding nValue(decimal) to Value Block value.

```
HF_RESULTRfid1356MifareIncrement (
    LPWSTRszBlock,
    LPWSTRszKeyType,
    LPWSTRszKeySet,
    uintnValue
);
```

Parameters

szBlock

designating block as 16 hexadecimal number. Ex) "00" or "01" or "0F"

szKeyType

access key. Ex) "A" or "B"

szKeySet

key set. Ex) "0" or "1", or "2"

nValue

Value to be added(decimal)

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Access is 1 sheet 1 block at one time, and Mifare Light cannot be added.(limitations of Card)

2.4.11 Rfid1356MifareDecrement

Subtracting nValue(decimal) from Value Block.

```
HF_RESULTRfid1356MifareDecrement (
    LPWSTRszBlock,
    LPWSTRszKeyType,
    LPWSTRszKeySet,
    uintnValue
);
```

Parameters

szBlock

designating block as 16 hexadecimal number. Ex) "00" or "01" or "0F"

szKeyType

access key. Ex) "A" or "B"

szKeySet

key set. Ex) "0" or "1", or "2"

nValue

Value to be subtracted(decimal)

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Access is 1 sheet 1 block at one time,

2.4.12 Rfid1356MifareUltraReadBlock

Reading block of MifareUltra Light Tag within the range of response

```
HF_RESULTRfid1356MifareUltraReadBlock (
    LPWSTRszTagNum,
    LPWSTRszBlock,
    LPWSTRszReadNum,
    LPWSTRszReadData
);
```

Parameters

szTagNum

Turn of tag to read. Must input "01"

szBlock

designating block as 16 hexadecimal number. Ex) "00" or "01" or "0F"

szReadNum

Digits of data to read

nReadData

variable that will store block value

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

there will be fail answer from block number 16 on, and reading blocks are limited to a maximum of 16. sum of The Block Number and the number of block are limited to 16.

2.4.13 Rfid1356MifareUltraWriteBlock

Reading block of MifareUltra Light Tag within within the range of response

```
HF_RESULTRfid1356MifareUltraWriteBlock (
    LPWSTRszTagNum,
    LPWSTRszBlock,
    LPWSTRszWriteData
);
```

Parameters

szTagNum

Turn of tag to read. Must input "01"

szBlock

designating block as 16 hexadecimal number.Ex) "00" or "01" or "0F"

szWriteData

Data which will be stored in block.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

The block number "00", "01" are the section that writing is forbidden(write-protection)

2.4.14 Rfid1356Iso15693ReadUid

Reading UID of ISO15693 Tag within the range of response.

```
HF_RESULTRfid1356Iso15693ReadUid (
    LPWSTRszUid
);
```

Parameters

szUid

variable where UID will be stored.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

2.4.15 Rfid1356Iso15693ReadBlock

Reading data of designated block of ISO15693 Tag within the range of response

```
HF_RESULTRfid1356Iso15693ReadBlock (
    LPWSTRszBlock,
    LPWSTRszReadCount,
    LPWSTRszReadData
);
```

Parameters

szBlock

Designate block to read as hexadecimal number. Ex) "00" or "01" or "0F"

szReadCount

Designate the number of block to read as hexadecimal Ex) "00" or "01" or "0F"

szReadData

variable where block data will be stored.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

2.4.16 Rfid1356Iso15693WriteBlock

Recording data to designated block of ISO15693 Tag within the range of response

```
HF_RESULTRfid1356Iso15693ReadBlock (
    LPWSTRszBlock,
    LPWSTRszWriteData
);
```

Parameters

szBlock

Designate block to read as hexadecimal number. Ex) "00" or "01" or "0F"

szWriteData

Data which will record block.

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Section which can record data are User Data, AFI and DSFID.

2.4.17 Rfid1356Iso15693LockBlock

Apply write-protection to designated block of ISO15693 Tag within the range of response.

```
HF_RESULTRfid1356Iso15693ReadBlock (
    LPWSTRszBlock
);
```

Parameters

szBlock

Designating block to read as hexadecimal number. Ex) "00" or "01" or "0F"

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Data can lock 1 block(4Byte) at a time, once locked block cannot be reverted to writable status.

2.4.18 Rfid1356Iso15693ReadAfi

Reading UID of Tag at the same AFI(Application Family Identifier) status among ISO15693 Tag within the range of response

```
HF_RESULTRfid1356Iso15693ReadAfi (
    LPWSTRszData,
    LPWSTRszAfi
);
```

Parameters

szData

variable for saving UID

szAfi

Application family identifier

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

Notes

AFI 1byte is composed of top 4bit of Family and sub 4bit of sub-Family.

If Sub-Family is 0, matching the Family is detected, and if both Family and sub-Family are 0, all tag are detected. Depending on type of tag, if family and sub-Family doesn't coincide, there is possibility of non-responding

2.4.19 Rfid1356Iso15693WriteAfi

Changing AFI(Application Family Identifier) registered in Tag

```
HF_RESULTRfid1356Iso15693WriteAfi (
    LPWSTRszAfi
);
```

Parameters

szAfi

AFI to be registered in Tag(hexadecimal)

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

2.4.20 Rfid1356Iso15693LockAfi

Apply write-protection to AFI(Application Family Identifier) registered in Tag

```
HF_RESULTRfid1356Iso15693LockAfi();
```

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully

Notes

Once locked AFI cannot be reverted to writable status.

2.4.21 Rfid1356Iso15693WriteDsfid

Changing DSFID(Data Storage Format Identifier) registered in Tag

```
HF_RESULT Rfid1356Iso15693WriteDsfid(  
    LPWSTR szDsfid  
);
```

Parameters

szDsfid

DSFID to be registered in Tag(hexadecimal)

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

2.4.22 Rfid1356Iso15693LockDsfid

Applying write-protection to DSFID(Data Storage Format Identifier) registered in Tag.

```
HF_RESULT Rfid1356Iso15693LockDsfid();
```

Parameters

None

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.

Notes

Once locked DSFID cannot be reverted to writable stuff

2.4.23 Rfid1356Iso14443APDU

Transmitting ISO 14443A/B APDU(application protocol data unit)

```
HF_RESULT Rfid1356Iso14443APDU(  
    LPWSTR szSendData,
```

`LPWSTR`szResultData

);

Parameters

szSendData

APDU command which will be transmitted to Rfid Module.

szResultData

The result about transmitted APDU command

Return Values

HF_RESULT_SUCCESS will be returned, if performed successfully.